REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 9/25/05	3. REPORT TYPE AND DATES COVERED Final Progress Report 7/01/02 -- 06/30/05

4. TITLE AND SUBTITLE
On Models for Coordination of Activity and its Disruption

5. FUNDING NUMBERS
DAAD19-02-1-0211

6. AUTHOR(S)
Joseph Lakey, Mike Coombs,
Scott Izu, Chris Weaver

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
New Mexico State University
Las Cruces, NM

8. PERFORMING ORGANIZATION REPORT NUMBER
AS02-0145
NMSU-2003005

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
43787.3-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The aim of this three year project under the contract number ARO DAAD19-02-1-0211 was to. develop a mathematical modeling and metrics framework for supporting the threat evaluation stage of the Intelligence Preparation of the Battlefield (IPB) process of asynchronous Low Intensity Conflict (LIC) threats (FM34-7, Section 3-9). The first goal was to assess the extent to which emergent coordination happens among agents in so-called ping system simulations that model action taking place in a constrained resource environment. The second, related goal, was to assess whether the simulation systems possess near optimal fitness states

This report discusses how these assessments were made in the contexts of specific sets of agent decision and action protocols..

14. SUBJECT TERMS complex adaptive systems agent systems scarce resource domains	15. NUMBER OF PAGES 24
	16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT **UNCLASSIFIED**	18. SECURITY CLASSIFICATION ON THIS PAGE **UNCLASSIFIED**	19. SECURITY CLASSIFICATION OF ABSTRACT **UNCLASSIFIED**	20. LIMITATION OF ABSTRACT **UL**

On Models for Coordination of Activity and its Disruption: Final Progress Report, 7/01/02 – 06/30/05

Dr. Joseph Lakey (PI), Department of Mathematical Sciences
Dr. Mike Coombs, Physical Science Laboratory
Scott Izu, Department of Mathematical Sciences
Chris Weaver, Physical Science Laboratory

New Mexico State University
Las Cruces, NM 88003
September 2005

Contents

List of Figures

List of Tables

1 Statement of Problems Studied

The stated aim of this three year project under the contract number ARO DAAD19-02-1-0211 was to develop mathematical principles for supporting the threat evaluation stage of the Intelligence Preparation of the Battlefield (IPB) process for asynchronous Low Intensity Conflict (LIC) threats (FM34-7, Section 3-9). Relevant Army programs include the ARL Collaborative Technology Alliance in Advanced Decision Architectures (CTA/ADA) managed by the U.S. Army Intelligence Center (USAIC) at Ft. Huachuca. This work provides theoretical support both for the design of secondary command and control (C^2) coordination tactics for distributed forces, and for the analysis of enemy tactics for IPB. Results of this research were reported at Ft. Huachuca in December, 2004.

Of particular interest was the capability of a multifactional force to pose a coordinated threat subject to environmental constraints. Example threats included multifactional insurgencies (*e.g.*, in Somalia – both clan-based (1990s) and more recently led by Islamic groups, ongoing ethnic conflicts in Kosovo, and coordinated suicide bombings in Iraq) or from an international terrorist group (*e.g.*, al-Qaeda). An important tenet in modeling such systems is that individual groups act *without* central guidance (though, perhaps, in accordance with general training doctrine), but groups can adapt their actions based on those of others. This raises an important limitation concerning the types of threats that we model. The attacks of 9/11 were scripted. They did not occur as a spontaneous response to indigent circumstances of an ongoing struggle and, therefore, do not fall into the category of insurgency to which our models pertain. Centrally coordinated or scripted threats require high level information and are best dispelled by tenaciously gathering, decoding, assembling, and analyzing that information.

In multifactional insurgencies, different groups often compete for common resources in their quest to pose a sustained threat to a common adversary. They are not prone to cooperate, or even to communicate plans. So the term *coordination* here requires a modicum of explanation: it refers to the ability of a collection of agents or factions to sustain conditions that enable a series of separate acts to be carried out and to achieve their intended outcomes. In this sense, the Iraqi insurgencies fit within our scheme but the actions of 9/11 do not.

The general objective of this project was to develop a *low level* modeling and metrics framework for such asymmetric threats. Specific goals included:

- prediction of the emergent coordination or disruption of activity in social systems models that
  - employ regulating feedback, often construed as violence, as a means of achieving goals and
  - use learning to improve performance;
- assessment of whether the observed systems possess near optimal fitness states that
  - can be attained quickly via a decentralized search strategy and
  - are robust to small perturbations of rule parameters or environment;
- determination of conditions favorable or opposed to such optimization.

In the 2003 and 2004 Interim Progress Reports for this program, the systems we sought to model were referred to as "communities of violence" and the notion of a 'ping' was used to symbolize violence. However, a ping could be interpreted in a broader sense as a form of symbolic threat that prevents another agent from taking a certain action. With this broader sense our models closely reflect other social systems that partially model control of territory such as traveling salesmen as outlined in [14]. While such generic models support general conclusions, one has to be certain that model parameters accurately map to physical systems of interest before one can infer behavior of the real system from simulations. Nevertheless, our mathematical framework certainly provides a means to quantify low-level coordination capability through performance metrics and, possibly, methods for influencing coordination.

1.1 Basic considerations for our models

Two key features separate our work from the bulk of the growing and already considerable research on *complex adaptive systems* (CAS) modeling (*e.g.*, http://www.casresearch.com/) and even on much of that relating to asymmetric threats, (*e.g.*, http://www.cna.org/isaac/terrorism_and_cas.htm). First, we aim ultimately to model human social systems, albeit at a low level. Secondly, we seek to model *meso-scale* systems – collections of agents containing more entities than what can manageably be analyzed by hand, but too few to justify analysis based on large scale averages. A collection of terrorist cells or a multi-factional insurgency involving 10 to 100 active units would fall in this category.

It is often noted that social networks display complex collective behavior despite simple interaction rules. It is often the case though, as with swarm models, (*e.g.*, Bonabeau et al., [4]) that individual, and even collective, behavior is *predictable*. Human behavior is much less so. Models of particular human social systems such as insurgencies will never be able to predict when and where a particular event will occur. At best, simulations can indicate preconditions for collective actions when hypotheses on the basic natures of interactions are met.

Additionally, human systems adapt in short time frames through communication of ideas – not through large risks for the purpose of incremental information gains as in the case of stable biological/ecological systems. So it does not often make sense to model change in a human social system through genetic search algorithms as one would do, say, for Kaufmann nets [11], *cf.* [10]. Nor is it prudent to model human social systems as efficiently designed systems of seamlessly interacting parts. Instead, we choose to model flows in social networks in terms of responses to and actions in response to an agent's social and material environment whose constraints induce competition for resources.

In addition to *decision* and *action* rules typical of any CAS model, a social network also employs some form of *communication*, *e.g.*, [6]. In our communities-of-violence models this communication takes the form of a ping which imposes a delay, or inaction, on its recipient.

With these principles in mind, we sought methods for predicting the emergence and stability of coordinated activity in constrained environment simulations. This requires quantification of rules for interaction, of environmental constraints, and of meaning and degree of competition and coordination.

1.2 Coordination versus cooperation

By *coordination*, we mean the ability of agents to align their internal states in such a way that the entire system reaches a high level of performance. In a constrained resource environment, *coordination* takes a meaning similar to that in noncooperative games (*e.g.*, [8, 7, 5]) and necessarily differs from *cooperation*, which refers to agents employing commonly agreed-upon methods, including contractual resource sharing, to achieve common interests. Cooperation can result in a stronger competitive force at the group level and is a common element of systems utilizing *collective intelligence* (*e.g.*, [16, 17]). But it was not our goal to model the development of collective agreements.

We were concerned, instead, with understanding general conditions under which a collection of *greedy* agents can find, within a large and complex configuration space, sequences of actions that result in common gains. Technically, we say that coordination occurs when the utility of a group of agents is increased by their carrying out a collective action in such a way that this collective utility exceeds an *expected* sum of individual utilities. This does not necessarily imply that each agent benefits equally or even that agents all act in a homogeneous way. A good illustration of this, in an even more basic context than ours, is the case of *minority games* [5, 10]. The simplest version is an (iterated) binary voting game in which the winners are those casting a minority vote (which is well defined when the number N of agents is odd). This game, by definition, is not cooperative. However, coordination is perfectly well defined in terms of agents employing individual strategies such that, on average, the minority is larger than if agents voted at random. In contrast, if two agents were allowed to communicate with one another, and sufficiently rational to notice that expected long term payoff is less than half, then the pair could guarantee an average score of half by mutually agreeing to alternate their votes both in time and with one another. This would be *cooperation*.

In multi-player, matrix payoff games, agents learn optimal individual strategies via hypothesis testing [8, 9]. An agent must, in effect, act contrary to its perceived short term interests in order that the system realize an adequate estimation of the payoff function, provided the system is not so complex that an agent cannot estimate a systemic response to its actions (*e.g.*, when N is small). In contrast, in the Ping systems outlined below, *greediness* of individual agents seems a prerequisite for coordination in that it adds some predictability of the actions of others. Consequently such systems can be driven into limit cycles, though performance along those depends on several behavioral parameters.

One of our main findings, in fact, is that the extent to which collective reward can be aligned with individual utility is largely a consequence of how *constrained* a system is: when the system has enough internal degrees of freedom, system performance is correlated, on average, with individual performance. In contrast, in highly constrained systems, collective performance tends to be optimized when the few degrees of freedom are exploited by a small subcollection of agents. The remaining agents become victims of their greed. Thus, while one must be careful in regarding the analysis of the particular simulations discussed below as indicative of general principles, nonetheless our simulations at least seem to be consistent with notions of ergodicity in the complex adaptive systems (CAS) literature.

2 Summary of Results

Most of the progress made under this program has already been reported in the interim progress reports [12, 13, 14]; some new observations are made at the end Section 2.2.3.

We sought to quantify essentially three aspects of performance in ping systems. First, how does contention among agents for access to external resources dictate protocols for movement and updating information? Secondly, how does physical geometry influence network connectivity and vice-versa? Thirdly, how can one predict system performance when dynamical rules are discrete and actions are based on nonlinear agent thresholds, prohibiting the application of classical tools of differential equations, but not necessarily of combinatorial tools?

The first question is addressed in the discussion of the particular model protocols for Ping I and Ping II as discussed in Sections 2.1.1 and 2.2.1. The second question is addressed, to some extent, by looking at the differences between Ping I and Ping II. Ping I is a linear model, having a single bottleneck. Ping II, in contrast, is a grid model. Congestion can form locally when the density of agents (number of agents over number of nodes) is high. We have not extended Ping II to agents moving on graphs other than periodic grids; however, we have tried to quantify performance metrics in such a way that the analysis can be extended readily to this important, more general case. In terms of metrics, we have quantified system *capacity* – a measure of optimal performance given system constraints. This provides a-priori bounds against which actual performance can be measured. It also provides a baseline for performance under different agent protocols, testing differences in performance for different parameters. More refined metrics include *efficiency*, which takes into account ability of a system to respond to constraints, and *proportionality of response*, which quantifies the extent to which agents do or do not form cycles of retribution in response to pressure brought on by system constraints. A basic discovery is that our Ping systems perform better, in general, when agents do not form such cycles. In terms of our prior remarks regarding coordination, this means that systems that are able to coordinate tend to be ones in which feedback is dispersed. One important question that we have not been able to address is that of determining precise preconditions, phrased combinatorially in terms of system configurations or short sequences thereof, under which coordination breaks down even though global constraints may not be so severe.

2.1 Coordination in a bottlenecked environment: Ping I

In Year 1 we developed and studied a simulation platform, that we called *Ping I*, for modeling communities-of-violence in which conflict arises over access to a common resource that is required intermittently by each faction (*e.g.*, access to roads or civic communications). Intermittency allows that, for certain ratios of supply and demand, factions can share the resource efficiently if their needs for access are desynchronized. Of principal concern is whether, and under what sort of interaction policies, is desychronization possible when the resource is *scarce* in the sense of being insufficient to satisfy all needs on a continual basis. *Violence* (pinging) is expressed as a reaction to scarcity and takes the nonlethal form of temporary physical obstruction. Ping I embodies specific elements of Rosenschein and Zlotkin's [15] generalized models for "restricted usage/scarce resource" domains. To quantify these issues in basic terms, we designed Ping I as a one-dimensional agent-based simulation platform in which agents were visualized as responding to frustration resulting from waiting for a *supply* resource required intermittently upon completing a sequence of *production steps*.

2.1.1 Outline of Ping I Mechanics

Ping I systems involve N agents $A_1, \ldots, A_N$ that individually and repeatedly complete process cycles of L steps. After the final process step each agent must complete an update/resupply step which involves accessing a shared resource having S access channels. The condition of having more demand than supply is quantified by the constraint that $C = S \cdot N/L < 1$: then there are fewer access channels than agents per step so it is impossible for the agents to desynchronize in such a manner that each agent has access when it completes its cycle.

In this case, an update queue forms and agents are frustrated by those who are in front. A_i's frustration takes the form of incrementing a weight w_{ij} when A_j precedes A_i. In each step, A_i checks its net frustration, Σw_{ij} over those A_j in front of A_i – either in terms of processing step or supply queue. If this sum exceeds a fixed threshold T, A_i *pings* a fixed number P of those agents that have frustrated A_i the most (with certain tiebreaking rules). The weight w_{ij} is also incremented whenever A_j pings A_i. For each ping received, A_i must remain stationary for one time step. The simulation parameters are summarized in Tables 1 and 2. Basic performance metrics are summarized in Table 3. Figure 1 contains a visual representation of agent states in which the update step is represented as access to a database. Figure 2 illustrates evolution of ping chains. Plots on left in that figure show ping activity is seeded by update frustration and mean system processing rate decreases as delays accumulate.

Table 1: Summary of Ping I dynamical steps for A_i

1	if threshold exceeded then ping
2	compute incoming pings
3	if in *processing mode*: if delayed then decrement delay counter else proceed forward one step if not delayed and processing completed then switch to *resupply* mode else switch to *queued* mode
4	if queued and channel open then switch to resupply mode else decrement queue counter by S
5	if in update mode then switch to *processing* step one
6	update weight counters

Table 2: Ping I system parameters

M	number of agents or model size
L	processing steps per cycle
S	source capacity
T	threshold
P	agent fanout – number of agents pinged and delayed

2.1.2 Conditions for coordination

Production capacity C characterizes the degree of constraint of the system. We will say that a Ping I system is *coordinated* provided the overall production $R(n)$ is close to system capacity C, at least for large enough n. General system behavior based on the ratio P/T is summarized in Table 4 and asymptotic mean fitness values are plotted in terms of (P, T) in Fig. 4.

To justify the observations in the table heuristically, suppose that up to S agents can be updated in each time step. If $C = SL/N < 1$ then, on average, some agents cannot be updated. Let d denote the average

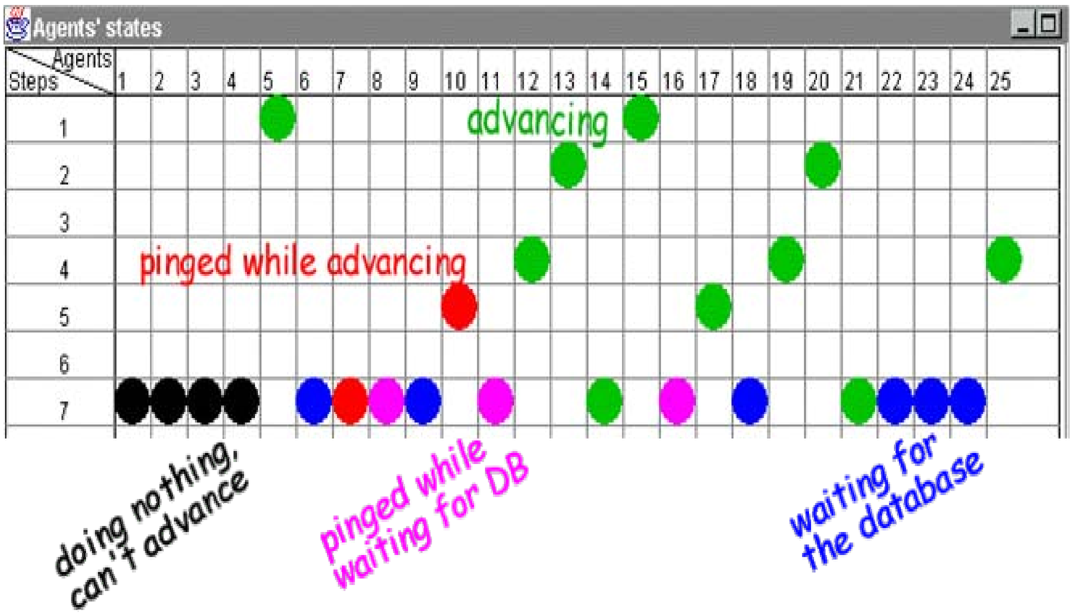

Figure 1: *Visual representation of states of a system of 25 agents with production cycle length $L = 7$.*

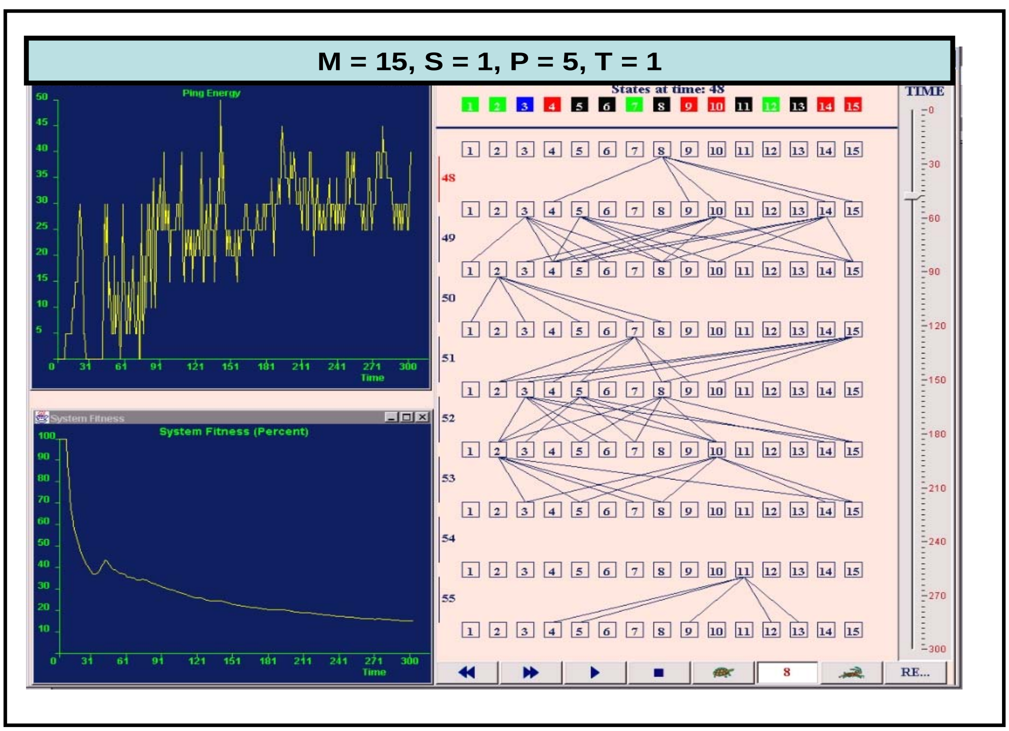

Figure 2: *Ping chains for a system of 15 agents over several consecutive steps.*

Table 3: Parameter dependent performance metrics for Ping I

Performance metric	mathematical notation
system production capacity	$C = SL/N$
total process steps by A_j to time n	$\mathrm{Pr_j(n)}$
mean process rate or *fitness* of A_j	$\mathrm{Pr_j(n)/n}$
mean system process rate or fitness	$R(n) = (\sum_j Pr_j(n))/(Nn)$

number of delays that each agent receives in a processing cycle. Thus, on average, agent process cycles take $L+d$ steps to complete. Then there should be Nd total pings distributed per $L+d$ times. In order that the agents can be desynchronized, that is, at most S agents complete their process cycles in each time step, we must have $S(L+d) > N$. In order that update space is not under utilized, we should then have $S(L+d) = N$ or $d = (N/S) - L$.

Since, over the system, the number of pings given equals the number of delays received, Nd/P agents should ping per $L+d$ steps, or on average, an agent should ping every $P(L+d)/d$ times. If the only source of frustration were to come from pings then an agent should receive T pings every $P(L+d)/d$ steps. Then $Td/(P(L+d)) = d/(L+d)$ or $T = P$. On a more intuitive level, if $P > T$ then delays accumulate in the system, while if $P < T$ then some agents will inevitably have to wait in the update queue.

While the condition $P \leq T$ is necessary for systems to realize full capacity, there is no guarantee that pinging will desynchronize the system. Conversely, in some situations agents may not ping until well after exceeding threshold. In this case we could still have a coordinated system with $P \gtrapprox T$. This behavior is borne out in simulations. Of course, in the limit $T \to \infty$ there is no pinging and capacity is always realized.

It is worth noting that, in our simulations, agents are allowed to ping while delayed. This results in a faster decline in fitness as T gets small in relation to P. Conversely, if agents were not allowed to ping while delayed then systems would be even more efficient.

Table 4: General performance and P/T

P/T	effect	long term behavior
$P/T \gg 1$	ping feedback, delays accumulate	frozen subsystems
$P/T \ll 1$	pings dissipate frustration	performance near capacity
$P/T \approx 1$	ping frustration balanced	coordination, intermittent feedback

2.1.3 Further indicators of performance in Ping I systems

Several other aspects of Ping I systems were analyzed and reported in the Interim Progress Report for Year 1 [12] to which we refer for detailed analysis of the following aspects of Ping I systems:

- Growth of configuration space as functions of:
  - Threshold T: possible weight configurations grow polynomially in T for fixed N
  - Number of agents N: both weight vector and ping matrix configurations grow exponentially in N for fixed threshold T and fanout P.
- Periodicity. Ping I systems settle into periodic limit cycles quickly when $P > T$, while dynamics remain *ergodic* when $P > T$.

Structure of limit cycles is an important aspect of these systems. As a general rule, feedback of ping energy among agents drives down the fitness of those agents. The complexity of system dynamics has precluded our finding precise expressions determining when such feedback is imminent in systems that are not highly constrained, though we have observed the emergence of this sort of feedback. Heuristically, this happens when many agents are close to the update step (in front of one another) and are *close to threshold* among all subsets of weights. In such cases, ping feedback loops can form when $P \approx T$ and drive the system to poor performance.

With these observations in mind we defined three metrics quantifying these correlates of fitness, namely

- *Proportionality of feedback*. This quantifies the extent to which the pings distributed by an agent correlate with the (ping) frustration received from those agents. A large value indicates feedback structures indicative, in turn, of unsatisfactory performance. A particularly relevant question, but one that is still open, is whether there is a threshold value for this metric beyond which feedback becomes self-sustaining at a high level systemwide (when $P > T$). Such a threshold would indicate particular vulnerability of a system to a perturbation (*e.g.*, external forcing) that would lead to poor global performance. Fig. 3 shows that, generally, feedback becomes more proportional as P/T increases.
- *Effective threshold*. For a given ratio P/T some systems can still perform better than others, largely depending on marginal ability to remove stored frustration through pinging. The effective threshold measures how much total frustration weight has accumulated, on average, before an agent pings. Recall that A_i only checks its threshold against its weights summed over those A_j *in front* of A_i. If A_i's weights tend to be evenly distributed over all agents then A_i will likely have a high effective threshold. Thus, this metric complements proportionality of feedback. When agents are desynchronized in their update cycles, systems with high effective threshold tend to be *efficient* in the sense that update channels tend to be utilized to their full extent.
- *Volatility* of a ping system can be expressed in terms of the distribution in time of ping events with large magnitudes. In well-performing systems the principal of *self-organized criticality* [2] stipulates that this distribution will obey a power law, that is, will be close to log linear. Volatility can then be expressed as deviation from log-linearity. Values of this volatility metric as they depend on P and T are plotted in Figure 5.

Further information regarding system complexity is contained in [14, 14]. Perhaps the most compelling question regarding Ping I systems is: are there simple rules under which agents can learn, given a fixed threshold value T, to adapt their fanout parameter for better overall performance? Secondly, are there simple rules under which agents can learn to self-impose a delay in order to avoid being delayed further? The difficulty with both of these questions boils down to complexity: if all agents are following more complex rules, then the problem of how to learn a best strategy in an unpredictable environment is highly ill posed.

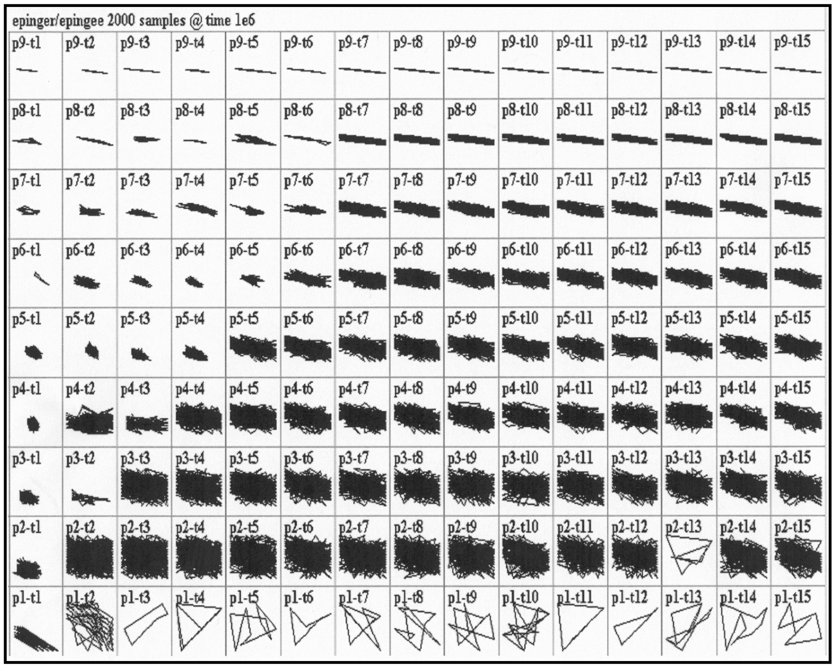

Figure 3: *Expected pinger-pingee dynamics*

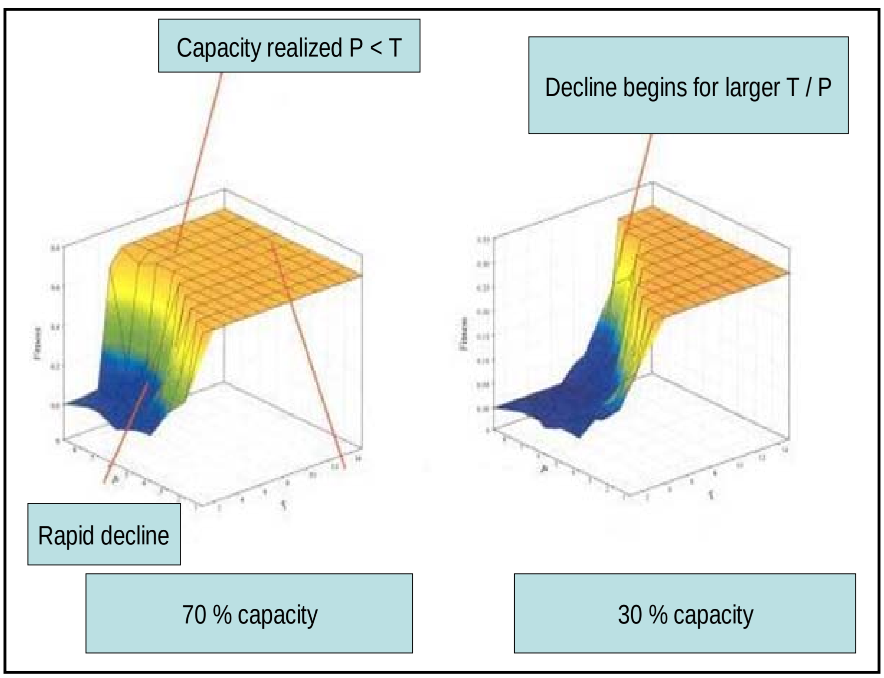

Figure 4: *Production versus capacity*

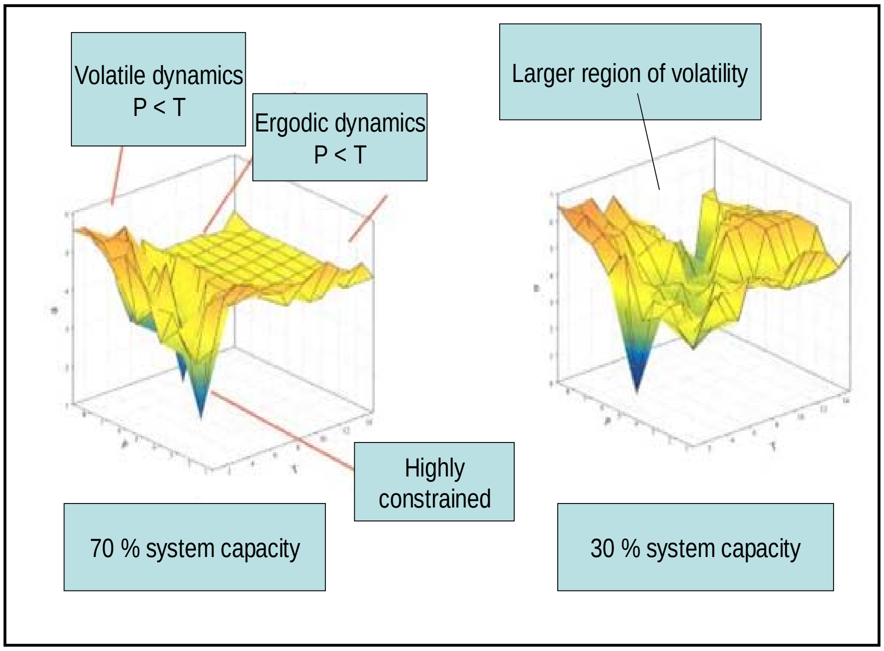

Figure 5: *Volatility as it depends on P and T and on capacity*

2.2 Coordination in collection of a distributed resource: Ping II

In Ping I the update step represents access to a shared resource. *Bandwidth constraints* amount to a bottleneck when $SL < N$ and performance of a Ping I system boils down, essentially, to the question of whether agents can desynchronize their need for access. In many domains, the resource is spatially distributed but still shared. Bottlenecks can still develop locally. In order to achieve reasonable global productive behavior, a system of agents then must be capable of searching collectively and implicitly for allocation schedules consistent with the constraints of the system. Assuming that agents are inherently greedy, *i.e.*, an agent will only act to maximize its short term gains, and will only forego reward when a threat is perceived, pinging in such a system can serve to offset agent greed, enabling the whole system to perform better.

In Year 2, we extended the one-dimensional model (Ping I) to a two-dimensional grid model (Ping II) in which agents seek to collect *reward* that is distributed and regenerated at grid nodes. In addition to retaining the aspects of greedy agents who can ping one another to enforce delays, we also endowed agents with low-level, neural network based learning of risk and reward relationships between its current position (in space and time) and configurations of neighboring agents.

With geometry comes additional complexity. For example, *relative* update capacity is somewhat analogous to agent density – the total number of agents divided by the total number of nodes; but it also depends on the rate at which reward is generated at nodes. Despite its added complexity, Ping II is only marginally more realistic as a model of agent behavior than is Ping I.

Ping II is an example of a *mobile social network* in which agents move along the edges of a graph according to some set of update rules. Reward is distributed at the nodes of the graph and is updated according to some allocation rule (see Section 2.2.2). Each individual agent seeks to collect reward at an optimal rate. Local update rules and reward allocation should conspire to provide satisfactory system performance. The two goals – local and global optimization – can be in conflict because, by collecting reward as rapidly as possible, an agent might diminish what remains for other agents in its wake. Thus we see an interplay of four key features of a social network at this level of abstraction:

- Topology of the underlying graph on which reward is allocated and collected

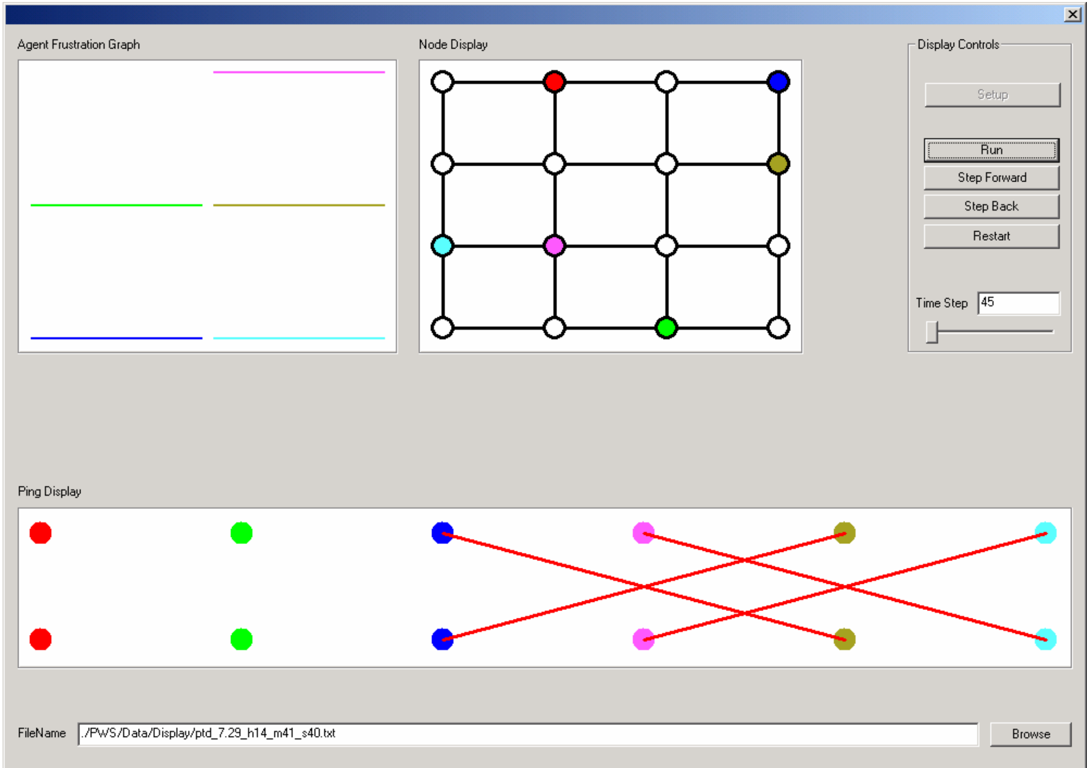

Figure 6: *Screenshot of Ping II visual display window*

- Law(s) by which reward gets allocated
- How agents interact with one another
- Agent decision rules for where to move next

In order to keep topological issues reasonably simple, we work specifically with square toral (*i.e. periodic boundary*) grid models in which agents can move in one of four directions (N,S,E,W). One important feature of grid geometry is scalability: basic quantities like *density* (agents per node) are independent of system size.

Interaction is the hallmark of a social system. As Ferber [6] suggested, *tags* play a fundamental role in communication networks. The simplest type of tag at a site just keeps track of when the site was last visited and/or who last visited the site. We use this same information for allocation of reward. One of the most significant aspects of Ping II as a mobile social network is that an agent can take an action to prohibit another from collecting reward, under certain conditions. Coupled with the law for allocation of reward, when rational agents are able to form appropriate hypotheses regarding the consequences of their actions in light of this form of coercion, it is possible that a Ping II network can exhibit a distribution of reward collection that is consistent with a reasonable notion of coordination – namely that agents (i) avoid getting in each others way – insofar as this is possible and (ii) by doing so, global achievement of reward can – in principle – be optimized when individual reward is distributed in a homogeneous way, *i.e.*, coordination can occur.

2.2.1 Outline of Ping II Mechanics

Figures 6, 7 and 8 show screenshots of Ping II simulation windows for entering simulation parameters, and for visualizing and analyzing agent interactions. The simulation environment was developed in visual C++ first by Chris Weaver, and further by Scott Izu and Michael Eydenberg, all PhD students at NMSU.

The Ping II display window shown in Fig. 6 provides a visual illustration of evolving agent positions, ping interactions, and frustration levels. Agents are represented by different colors. Interactive features include ability to step forward and back up, thus visualizing sequences of agent movements and interactions. Display files output from different simulation runs can be loaded into the window, thus allowing for direct visual comparison of different simulations.

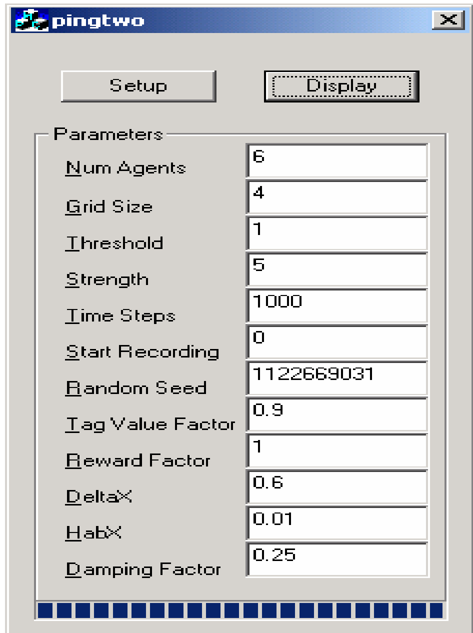

Figure 7: *Screen shot of Ping II setup window.*

The setup window in Fig. 7 for Ping II simulations allows the user to specify all of the system parameters specified in Table 6, in addition to neural net parameters.

The simulation data window in Fig. 8 shows a file that keeps a record for the main data for each time set of a Ping II simulation, including agent positions, delays, and accumulated reward. Alternate versions output fuller data, including frustration matrices and ping matrices.

Agent logic. In Ping II, a system of N agents inhabits a two-dimensional square grid of size $K \times K$. Each agent is assigned an identification number $i \in \{1, \ldots, N\}$. As before, we refer to the i-th agent as 'A_i'. Like Ping I, Ping II is a *threshold-delay* model. Simulations are initialized with a 'pseudo-random' seed that determines initial configurations of all agents.

In each time step the agents undergo a two-stage process – *action* then *update* – taking their turns in each stage according to a (randomly initialized) queue. Those agents that are most delayed go to the back of the queue. Agents that get to move first in a time step have first opportunity to gather reward that is within reach. For the purpose of this report, agents are *homogeneous* in that threshold and fanout parameters are the same for all agents – as was the case for Ping I. The agent logic in each stage is summarized in Table 5.

Here are some brief comments about assignment of system parameters and decision/action protocols for agents. They are taken from the perspective of the understanding that we now have of Ping II systems, including some understanding of system capacity, that we did not have when we first introduced Ping II. Suppose for the moment that we have a reasonable way of formulating the capacity C of a Ping II system, as we will do below. Then it is natural to ask: *what sorts of homogeneous greedy agent protocols can result in system coordination and, thus, near attainment of capacity?* Again, the assumption of greed is non-negotiable: an agent can only act in a manner consistent with maximizing its expectation of reward – possibly long term if it has sufficient rational power. Our approach to studying conditions for coordination boil down to combinations of fixed agent protocols and agent and system parameters that induce or inhibit coordination.

First, when its turn comes, A_i moves to the site among its four von-Neumann neighbors that maximizes a cost-benefit value predicted by a neural net, based on the local conditions up to two steps from the neighbor site in question. The neural net is described in [14]. It takes into account the frustration levels of the agents in this region and their relative move order. Agents experience frustration over *ownership* – meaning some perception of control of a node, which might be defined different ways. We chose a simple rule: A_i thinks

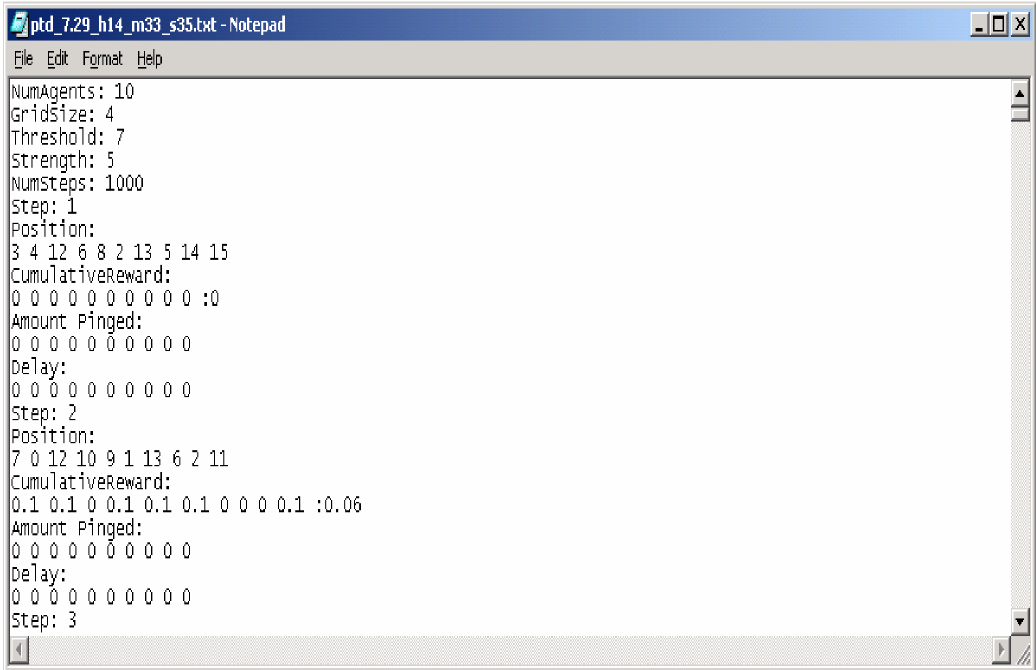

Figure 8: *Screen shot of Ping II simulation data window*

that it *owns* any site at which A_i has collected reward. A_i then increments its frustration weight for A_j by one unit whenever A_j collects reward at a site on A_i's ownership list. Other rules of perceived ownership are possible and we will mention them and their consequences in Section 2.2.3. A_i checks its frustration by summing *all* of its weights. If this sum exceeds a fixed threshold T then A_i distributes P pings as evenly as possible against any pingable neighbors – defined simply as those agents occupying one of the four nodes adjacent to A_i. If there are no such agents then A_i continues to be frustrated. Upon being pinged, A_i becomes passive for one time step for each ping received. This means that A_i can move, but cannot collect reward or ping. This is different from Ping I in which a delayed agent could ping but could not move. If A_i is pinged by A_j then it increments its weight for A_j by p_{ji} – the number of *ping units* received: here $\sum p_{ji} = P$.

With these simple rules (with further details as described in [14]) it is clear that, as was the case with Ping I, system behavior will depend on the relative parameters P and T. Capacity will also depend on the agent density defined as the number of agents per node. The situation is slightly different from Ping I, in which capacity depended on the number of agents, nodes, and update channels. Thus, in Ping II the source of initial frustration is different. The rule for pinging also has an important difference: in Ping II pings are directed at neighbors, not necessarily at those who pinged. This disproportionality can prevent formation of feedback of frustration. Finally, capacity also depends on the rules under which reward is allocated at grid nodes.

2.2.2 Reward and capacity

We have defined reward as a site specific quantity that is collected by agents as they visit nodes. The capacity and, intuitively, the ability of agents to coordinate depends on how reward accumulates at sites. We assume that when an agent visits a node it collects all the reward there. The basic assumption on reward is that it should be increasing between collection times. A less obvious axiom is that reward should be bounded. Not only is this consistent with real systems involving replenishable but locally bounded resources. It also prevents chaotic system behavior and allows us to make new and precise statements about the nature of coordination.

In view of these observations, we proposed to use a discrete logistic reward function, defined in terms

Table 5: Ping II agent logic

Stage 1: Update reward	If a neighboring site is open then:	Move to open site that maximizes: (Normalized Reward - Risk).
	If A_i believes that it controls its current node then:	check owner of current tag. If owner is A_j $(j \sim i)$ then increment w_{ij}: A_i's frustration counter for A_j.
	If delay is greater than zero then decrement delay counter. If delay is zero then:	Collect reward at site (one minus tag value). Reset tag value to $1/\beta$ and (if necessary) ownership to A_i.
Stage 2: Update delays	If A_i is frustrated (sum of frustrations exceeds threshold) and not delayed then:	Check for neighbors and distribute pings among them. Zero out frustration vector. Update risk neural net.
	Increment delay by number of pings received and increment w_{ij} for each ping from A_j.	

of a *tag factor* (see [14]) $\beta \in (0,1)$. Thus we define the reward $r(t)$ at a site t unit steps since it was last collected as $r(t+1) = r(t) + (1-\beta)\cdot(1-r(t)) = 1-\beta^{t-1}$ when $r(1)=0$, reflecting that one full time step must intervene before reward accumulates. For small β, $r(t) \uparrow 1$ quickly but $r(t) \uparrow 1$ gradually when $\beta \lesssim 1$.

With this replenishment function we derived the following general formula for system reward when agents A_i collect reward on *pairwise disjoint* sets C_i of collection nodes of magnitude $|C_i|$ and with d_i delays per cycle:

$$R(C_1,\ldots,C_n;d_1,\ldots,d_N) = \frac{1}{N}\sum_{i=1}^{N}\frac{|C_i|}{|C_i|+d_i}(1-\beta^{|C_i|+d_i-1}). \tag{1}$$

We were able to establish a general principle stating that, when the reward allocation function is concave, as it is in our case, system reward is maximized when $|C_i|+d_i$ are uniform across agents, with a *waiting time* value $t(\beta) = |C|+d$ that optimizes the reward function $R(\beta,t) = (1-\beta^{t-1})/t$. The optimal $t(\beta)$ is just over 5 when $\beta = 0.9$ and just under 2.5 when $\beta = 0.5$. The waiting time is the number of steps between successive reward collections at a site and must be a whole number. When the optimal waiting time is nonintegral the waiting times should average out to the optimal waiting time, with minimal variance. The problem then is to come up with a decomposition of the grid into disjoint subsets of nodes C_i, and agent closed paths $\widetilde{C}_i$ of length $|C_i|+d_i$, such that agents collect at nodes of C_i and are inactive at the other d_i nodes, in such a way that (i) $|C_i|+d_i = t(\beta)$, the optimal waiting time and $|C_i|$ and d_i vary as little as possible.

Depending on grid size K and density N/K, the incompatibility of these criteria may not allow for near optimal performance. A more general approach allows agents to share collection sites in such a way that the *length plus delay equals optimal waiting time* criterion is met essentially at all nodes, but this might need to happen over several traversals of the grid. Examples of how this works are given in [14].

Given these observations we can finally formulate the capacity of a grid of size K with reward factor β:

Table 6: Ping II system parameters

M	number of agents or model size
K^2	number of grid points
$D = M/K^2$	agent density
β	tag value factor
T	threshold
P	agent fanout – number of delays distributed among neighbors

Table 7: Parameter dependent performance metrics for Ping II

Performance metric	mathematical notation
System capacity	*see below*
Agent reward to time n	$R_i(n)$ *see below*
Time average agent reward	$R_i(n)/n$
Average system reward	$R_i(n)/(nN)$

The capacity $C(K, \beta)$ of a grid of size K and fixed reward factor β is $C(K, \beta) = KR(\beta, t(\beta))$ where $t(\beta)$ is the unique maximizer of $R(\beta, t)$.

Given this definition of capacity, some limitations on the ability of a collection of agents to coordinate can be given right of the bat. First, if $K/N = t(\beta)$ and if the grid can be divided into N disjoint cycles each of length K/N then capacity can be realized with no delays. We then say that the agent density is β-critical, or simply critical. If the density is subcritical, that is, there are too few agents, then system capacity cannot be realized: the agents cannot cover the grid fast enough. If agent density is supercritical then delays are necessary in order to realize capacity.

It is difficult to give a precise formula for the desired relationship between threshold T and fanout P in order that a supercritical collection of agents will perform near capacity. Arguing as in the case of Ping I, one *can* formulate what the desired ratio P/T would be to realize the optimal waiting time $t(\beta)$ *on average* if agents moved randomly to a neighboring site with equal probability for each available node. However, in this case there would be nontrivial variance in waiting times about the optimal average $t(\beta)$ and the inevitability of significant deviations in the case of random behavior precludes system reward nearly attaining capacity.

In [14] we analyzed average and variance of waiting times as they depend on the parameters P and T. The essential finding was that, for a given replenishment factor β and supercritical density, systems that achieve a desired average waiting time are subject to wide variations of the same, while systems exhibiting small variations in waiting times either have too much or too little average delay. These results pertain to agents running under the protocols in Table 6. Thus, either more efficient action protocols or greater rational powers are required of the agents in order to (nearly) achieve capacity.

2.2.3 New observations on Ping II

In this final section we will report some new simulation analysis for Ping II that was done after submitting the 2004 Interim Progress Report. These results concern the performance of ping systems under alternative protocols for pinging. In each case the ping system is equipped with a neural net adjusted to system protocols. The differences between the three versions that we consider are summarized in the Table 8.

In version 1, a fixed number P of pings is divided among all pingable neighbors. In version 2, the fanout P is still fixed but only active neighbors are pinged. In particular, if all of A_i's neighbors are already delayed then A_i will not ping. In versions 3 and 4 the fanout is adapted to the frustration level (sum of weights) and the number of pings distributed among neighbors is this sum, minus the agent's threshold. Versions 3 and 4 differ in ownership rules. An agent places a tag at each node that it visits while in active mode, superseding any tags placed by previous agents. In version 3, A_i does not own a node until a situation arises that no other agent tags the node between subsequent visits by A_i. In version 4, A_i *owns* any node that it tags. Once A_i owns a node, it adds one unit of weight to its frustration any time another agent tags a node that A_i owns. As we have set things up, pinging resets frustration to zero but it does not eliminate ownership. It would be of interest to study the case in which a pinging agent relinquishes ownership at the least recently visited site: Recall that Ping systems are assumed to perform well when agents traverse disjoint collection cycles. But if every agent *owns* every node then there is no way to distinguish those nodes whose traversal will or will not frustrate other agents. The upshot of this observation is that there are several sensible sets of agent protocols that one can investigate; it takes some time to formulate hypotheses about such protocols, program and simulate the corresponding systems, and analyze their outputs.

In Tables 9 and 10 we report the total rewards, averaged over agents, for the different versions. In each case, the initial seed value is fixed and rewards are computed for different values of the parameters P and T. In Table 11 we consider the effect of different initial seed values for version 4.

Finally we consider the effect of using a different number of agents. In Table 12 we consider $N = 10$ agents. For comparison, in Table 13 the rewards for Version 4 with 10 agents.

Table 8: Different versions of Ping II

Version	Distinguishing Feature 1	Distinguishing Feature 2
1	P fixed pings	all neighbors pinged
2	P fixed	delayed agents not pinged
3	$P =$ frustration $-$ threshold	strong ownership
4	$P =$ frustration $-$ threshold	weak ownership

Table 9: Total average reward for P, T parameters, versions 1 and 2

$\beta = 0.9$, 4×4 grid, 6 agents, 10^4 steps. Top: version 1; bottom: version 2; initial seed: 1120000000

P T	1	2	3	4	5	6	7	8	9	10
1	1688	1672	1640	1615	1589	1611	1601	1601	1599	1600
	1704	1670	1629	1615	1589	1611	1601	1601	1599	1600
2	1617	1695	1690	1666	1637	1603	1609	1600	1599	1600
	1313	1698	1699	1668	1616	1620	1602	1602	1599	1600
3	1474	1441	1675	1651	1663	1638	1589	1603	1602	1606
	1452	1448	1672	1673	1666	1628	1603	1587	1597	1603
4	1462	1476	1532	1551	1623	1644	1650	1637	1628	1580
	1328	1219	1508	1522	1690	1654	1608	1618	1592	1587
5	1262	1639	1562	1452	1634	1662	1673	1667	1663	1598
	1415	1403	1462	1629	1672	1670	1669	1610	1664	1613
5	1408	1450	1651	1660	1586	1614	1640	1676	1609	1645
	1431	1245	1323	1503	1589	1603	1686	1678	1668	1652
7	1417	1550	1619	1452	1516	1642	1638	1690	1677	1602
	1223	1309	1555	1464	1596	1658	1685	1676	1640	1613
8	1402	1461	1683	1572	1565	1530	1589	1610	1664	1605
	1077	1590	1374	1640	1526	1616	1658	1678	1680	1667
9	1249	1473	1637	1530	1507	1477	1522	1560	1670	1674
	1323	1224	1291	1213	1678	1530	1635	1664	1683	1690
10	1080	1542	1599	1482	1574	1669	1642	1610	1568	1662
	1242	1175	1278	1224	1220	1451	1587	1623	1664	1684

Table 10: Total average reward for P, T parameters, versions 3 and 4

$\beta = 0.9$, 4×4 grid, 6 agents, 10^4 steps. Top: version 3; bottom: version 4; initial seed: 1121100000

T	1	2	3	4	5	6	7	8	9	10
1	802	965	951	951	951	951	951	951	951	951
	906	1428	1427	1412	1016	1366	1523	1555	1565	1541

Table 11: Effect of initial seeds on total average reward for T parameters

$\beta = 0.9$, 4×4 grid, 6 agents, 10^4 steps. Version 4;

seed $/T$	1	2	3	4	5	6	7	8	9	10
1	909	1466	1479	1472	1277	1453	1558	1452	1445	1450
10	745	1361	1478	1470	1485	1087	1560	1565	1484	1063
100	991	1005	1452	1417	1397	1381	1080	1554	1014	1445
1000	936	1439	1278	1412	1179	1536	1355	1196	1357	1381

Table 12: Reward for P, T parameters, 10 agents, version 1

$\beta = 0.9$, 10 agents; 4×4 grid; 10^4 steps, pseudo random seed 1120673491

P/T	1	2	3	4	5	6	7	8	9	10
1	6012	7333	6237	6012	6007	6005	6001	5999	5999	5999
2	9041	9322	8109	6104	6083	6006	6004	5999	5999	5999
3	9489	9594	9078	8354	6068	5999	6047	5999	5999	5999
4	9491	9599	9407	8978	7424	6080	6081	5999	5999	5999
5	9482	9576	9495	9237	8870	8456	6055	5999	5999	5999
6	9418	9518	9537	9240	9138	8813	8479	5999	5999	5999
7	9429	9535	9550	9464	9281	9055	8784	5999	5999	5999
8	9383	9527	9539	9502	9379	9210	8997	5999	5999	5999
9	9319	9494	9499	9503	9425	9314	9128	5999	5999	5999
10	9296	9525	9522	9528	9477	9380	92310	5999	5999	5999

Table 13: Reward for P, T parameters, 10 agents, version 4

$\beta = 0.9$, 10 agents; 4×4 grid; 10^4 steps, pseudo random seed 1120520563; Version 4

T	1	2	3	4	5	6	7	8	9	10
	1154	9538	8625	7983	7539	7245	7044	6893	6780	6689

3 Conclusions

We have described low-level mathematical models and corresponding simulation platforms for coordination of agents in constrained systems. The constraint is described in terms of a bottleneck in Ping I and in terms of a distributed, replenishable resource on a grid in Ping II. Despite being low-level, the simulations describe behavior of complex systems that cannot be captured and analyzed through closed form expressions. In both Ping I and Ping II, agents respond to pressures caused by mismatch between internal drive to complete tasks, versus environmental constraints that hinder these tasks, by imposing delays upon one another. In some cases these delays accumulate and drive the system into low productive states, while in others, delays balance out and do not prevent the system from achieving its capacity. We defined several metrics that facilitate analysis of performance based on combinations of system parameters and agent protocols. We have thus achieved some explanation of why and when some systems are able to coordinate and others not, in terms of internal agent parameters versus system constraints. In foreseen applications of these methods to threat evaluation, global constraints correspond to limited materials or communications channels, while internal parameters correspond to psychological variables (*e.g.* frustration) and perception of threat or position in a hierarchy relative to others requiring the same resources. Our results are still not complete in a number of important ways. First, we have modeled only homogeneous agents and environmental resources: we have not considered nonstationary situations in which environmental pressures depend on time or location, or in which some agents have higher strength (in terms of ability to affect others) or rational powers. Perhaps even more importantly, we have not yet been able to quantify in any automatically detectable way local preconditions for poor performance.

4 Publications and reports supported under this contract

Papers published in peer-reviewed journals

[1.] J.A. Hogan and J. Lakey, Sampling and Oversampling in shift-invariant and multiresolution spaces I: validation of sampling schemes, Int. J. Wavelets and Multiscale Inf. Proc., **3**, (2005), 257–282.

[2.] S. Efromovich, J. Lakey, M.C. Pereyra and N. Tymes, Data-driven and optimal denoising of a signal and recovery of its derivative using multiwavelets; IEEE Trans. Sig. Proc. **52**, (2004), 628–635.

Papers published in non-per-reviewed journals or conference proceedings

J.A. Hogan and J. Lakey, Periodic nonuniform sampling in shift-invariant spaces, *in* "Harmonic Analysis and Applications: In Honor of John J. Benedetto," C. Heil ed., Birkhäuser, Boston, to appear (invited chapter).

Papers presented at meetings, but not published in conference proceedings

None

Manuscripts submitted, but not published

J. Lakey, D. Birx, M. Coombs, S. Izu and C. Weaver, On models of coordination activity in communities of violence, submitted to Military Operations Research Journal, September, 2005.

Technical reports submitted to ARO

J. Lakey, M. Coombs, K. Streander and C. Weaver, *On models for coordination activity and its disruption: interim progress report, 06/01/02– 12/31/02.*

J. Lakey, M. Coombs, S. Izu and C. Weaver, *On models for coordination activity and its disruption: interim progress report, 01/01/03 – 12/31/03.*

J. Lakey, M. Coombs, S. Izu and C. Weaver, *On models for coordination activity and its disruption: interim progress report, 01/01/04 – 7/31/04.*

5 List of all participating scientific personnel

Dr. Joseph Lakey, Mathematical Sciences, New Mexico State University (PI)

Dr. Michael Coombs, Physical Science Laboratory, New Mexico State University (senior personnel)

Christopher Weaver, Ph.D. student. Mathematical Sciences, New Mexico State University

Scott Izu, Ph.D. student. Mathematical Sciences, New Mexico State University

Michael Eydenberg, Ph.D. student. Mathematical Sciences, New Mexico State University

6 Report of inventions

No inventions or subcontracts to report.

References

[1] J. Arquilla and D. Ronfeldt (Eds.), "Networks and Netwars: The Future of Terror, Crime, and Militancy." RAND Corp. National Security Research Division, 2001.

[2] P. Bak, "How Nature Works: The Science of Self-organized Criticality," Copernicus: New York, NY, 1996.

[3] A. Bechara, A. Damasio and A.R. Damasio, *Emotion, decision making and the orbitofronral cortex*, Cerebral Cortex **10**, (2000), 295–307.

[4] E. Bonabeau and G. Théraulaz, *Swarm smarts*, Scientific American March 2000, 73–79.

[5] G. Bottazzi, G. Devetag and G.Dosi, *Adaptive learning and emergent coordination in minority games*, Simulation Modeling Practice and Theory **10**, (2002), 321–347.

[6] J. Ferber, "Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence," Addison Wesley: Harlow, UK, 1999.

[7] D. Foster and H. P. Young, *On the impossibility of predicting the behavior of rational agents* PNAS **98** (2001), 12848–12853 `http://gosset.wharton.upenn.edu/ foster/research/`.

[8] D. Foster and H. P. Young, *Learning, Hypothesis Testing, and Nash Equilibrium*, Games and Economic Behavior **45** (2003), 73–96 `www.santafe.edu/sfi/publications/wpabstract/200302007`.

[9] D. Fudenberg and D. K. Levine, "The Theory of Learning in Games," MIT Press, 1998.

[10] A. Galstyan and K. Lerman, *Minority games and distributed coordination in non-stationary environments* in "Proc. International Joint Conf. on Neural Networks," May 2002, Hawaii.

[11] Kauffman, S.A., "The Origins of Order: Self-organization and Selection in Evolution." Oxford University Press: Oxford, UK, 1993.

[12] J. Lakey, M. Coombs, K. Streander and C. Weaver, *On models for coordination activity and its disruption: interim progress report, 06/01/02– 12/31/02.*

[13] J. Lakey, M. Coombs, S. Izu and C. Weaver, *On models for coordination activity and its disruption: interim progress report, 01/01/03 – 12/31/03.*

[14] J. Lakey, M. Coombs, S. Izu and C. Weaver, *On models for coordination activity and its disruption: interim progress report, 01/01/04 – 7/31/04.*

[15] J.S. Rosenschein and G. Zlotkin, "Rules of Encounter: Designing Conventions for Automated Negotiations among Computers." MIT Press, Cambridge, MA., 1994.

[16] D. Wolpert and Lawson, *Designing Agent Collectives for Systems with Markovian Dynamics*, in "Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3, July 15-19, 2002, Bologna, Italy," 2002.

[17] D. Wolpert and K. Tumar, *An introduction to collective intelligence*, in "Handbook of Agent Technology," J.M. Bradshaw, ed., AAAI Press/MIT Press, 1999.